Mining Contextual Information Beyond Image for Semantic Segmentation

Zhenchao Jin[1]*, Tao Gong[1], Dongdong Yu[2], Qi Chu[1]†,
Jian Wang[2], Changhu Wang[2], Jie Shao[2]
[1]University of Science and Technology of China [2]ByteDance
{blwx@mail., gt950513@mail., qchu@}ustc.edu.cn
{yudongdong, wangjian.pierre, wangchanghu, Shaojie.mail}@bytedance.com,

Abstract

This paper studies the context aggregation problem in semantic image segmentation. The existing researches focus on improving the pixel representations by aggregating the contextual information within individual images. Though impressive, these methods neglect the significance of the representations of the pixels of the corresponding class beyond the input image. To address this, this paper proposes to mine the contextual information beyond individual images to further augment the pixel representations. We first set up a feature memory module, which is updated dynamically during training, to store the dataset-level representations of various categories. Then, we learn class probability distribution of each pixel representation under the supervision of the ground-truth segmentation. At last, the representation of each pixel is augmented by aggregating the dataset-level representations based on the corresponding class probability distribution. Furthermore, by utilizing the stored dataset-level representations, we also propose a representation consistent learning strategy to make the classification head better address intra-class compactness and inter-class dispersion. The proposed method could be effortlessly incorporated into existing segmentation frameworks (e.g., FCN, PSPNet, OCRNet and DeepLabV3) and brings consistent performance improvements. Mining contextual information beyond image allows us to report state-of-the-art performance on various benchmarks: ADE20K, LIP, Cityscapes and COCO-Stuff [1].

1. Introduction

Semantic segmentation is a long-standing challenging task in computer vision, aiming to assign semantic labels to each pixel in an image accurately. This task is of broad interest for potential application of autonomous driving, medical diagnosing, robot sensing, to name a few. In recent years, deep neural networks [21, 36] have been the dominant solutions [1, 3, 5, 6, 8, 15, 39, 50], where the encoder-decoder architecture proposed in FCN [32] is the cornerstone of these methods. Specifically, these studies include adopting graphical models or cascade structure to further refine the segmentation results [4, 10, 10, 53], designing novel backbone networks [38, 44, 49] to extract more effective feature representations, aggregating reasonable contextual information to augment pixel representations [5, 39, 50] that is also the interest of this paper, and so on.

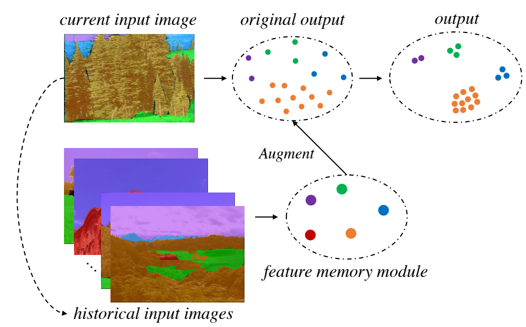

Figure 1. Mining contextual information beyond image (MCIBI). The *feature memory module* stores the dataset-level representations of various classes. The dotted line denotes that the *current input image* will be added into the *historical input images* after each iteration during training.

Since there exist co-occurrent visual patterns, modeling context is universal and essential for semantic image segmentation. Prior to this paper, the context of a pixel typically refers to a set of other pixels in the input image, *e.g.*, the surrounding pixels. Specifically, PSPNet [50] utilizes pyramid spatial pooling to aggregate contextual information. DeepLab [5, 6, 8] exploits the contextual information by introducing the atrous spatial pyramid pooling (ASPP). OCRNet [45] proposes to improve the representation of each pixel by weighted aggregating the object region representations. Several recent works [15, 22, 39, 46] first calculate the relations between a pixel and its contextual pixels,

*This work was performed while Zhenchao Jin worked as an intern at ByteDance.

†Corresponding author.

[1]Our code will be available at https://github.com/CharlesPikachu/mcibi.

and then aggregate the representations of the contextual pixels with higher weights for similar representations. Apart from these methods, some studies [1, 23, 52] also find that the pixel-wise cross entropy loss fundamentally lacks the spatial discrimination power and thereby, propose to verify segmentation structures directly by designing context-aware optimization objectives. Nonetheless, all of the existing approaches only model context within individual images, while discarding the potential contextual information beyond the input image. *Basically, the purpose of semantic segmentation based on deep neural networks essentially groups the representations of the pixels of the whole dataset in a non-linear embedding space.* Consequently, to classify a pixel accurately, the semantic information of the corresponding class in the other images also makes sense.

To overcome the aforementioned limitation, this paper proposes to mine the contextual information beyond the input image so that the pixel representations could be further improved. As illustrated in Figure 1, we first set up a feature memory module to store the dataset-level representations of various categories by leveraging the historical input images during training. Then, we predict the class probability distribution of the pixel representations in the current input image, where the distribution is under the supervision of the ground-truth segmentation. At last, each pixel representation is augmented by weighted aggregating the dataset-level representations where the weight is determined by the corresponding class probability distribution. Furthermore, to address intra-class compactness and inter-class dispersion from the perspective of the whole dataset more explicitly, a representation consistent learning strategy is designed to make the classification head learn to classify both 1) the dataset-level representations of various categories and 2) the pixel representations of the current input image.

In a nutshell, our main contributions could be summarized as:

- To the best of our knowledge, this paper first explores the approach of mining contextual information beyond the input image to further boost the performance of semantic image segmentation.
- A simple yet effective feature memory module is designed for storing the dataset-level representations of various categories. Based on this module, the contextual information beyond the input image could be aggregated by adopting the proposed dataset-level context aggregation scheme, so that these contexts can further enhance the representation capability of the original pixel representations.
- A novel representation consistent learning strategy is designed to better address intra-class compactness and inter-class dispersion.

The proposed method can be seamlessly incorporated into existing segmentation networks (*e.g.*, FCN [32], PSPNet [50], OCRNet [45] and DeepLabV3 [6]) and brings consistent improvements. The improvements demonstrate the effectiveness of mining contextual information beyond the input image in the semantic segmentation task. Furthermore, introducing the dataset-level contextual information allows this paper to report state-of-the-art intersection-over-union segmentation scores on various benchmarks: ADE20K, LIP, Cityscapes and COCO-Stuff. We expect this work to present a novel perspective for addressing the context aggregation problem in semantic image segmentation.

2. Related Work

Semantic Segmentation. The last years have seen great development of semantic image segmentation based on deep neural networks [21, 36]. FCN [32] is the first approach to utilize fully convolutional network for semantic segmentation. Later, many efforts [3, 5, 6, 8, 45, 50] have been made to further improve FCN. To refine the coarse predictions of FCN, some researchers propose to adopt graphical models such as CRF [4, 31, 53] and region growing [13]. CascadePSP [10] refines the segmentation results by introducing a general cascade segmentation refinement model. Some novel backbone networks [38, 44, 49] have also been designed to extract more effective semantic information. Feature pyramid methods [8, 24, 30, 50], image pyramid methods [7, 27, 28], dilated convolutions [5, 6, 8] and context-aware optimization [1, 23, 52] are four popular paradigms to address the limited receptive field/context modeling problem in FCN. This paper focuses on the context modeling problem in FCN.

Context Scheme. Introducing contextual information aggregation scheme is a common practice to augment the pixel representations in semantic segmentation networks. To obtain multi-scale contextual information, PSPNet [50] performs spatial pooling at several grid scales, while DeepLab [5, 6, 8] proposes to adopt pyramid dilated convolutions with different dilation rates. Based on DeepLab, DenseASPP [42] densifies the dilated rates to conver larger scale ranges. Recently, inspired by the self-attention scheme [37], many studies [3, 15, 39, 46, 51, 57] propose to augment one pixel representation by weighted aggregating all the pixel representations in the input image, where the weights are determined by the similarities between the pixel representations. Apart from this, some researches [26, 45, 47] propose to first group the pixels into a set of regions, and then improve the pixel representations by weighted aggregating these region representations, where the weights are the context relations between the pixel representations and region representations. Despite impressive, all existing methods only exploit the contextual information within individual images. Different from previous works, this paper proposes to mine more contextual information for each pixel from the whole dataset to further improve the pixel representations.

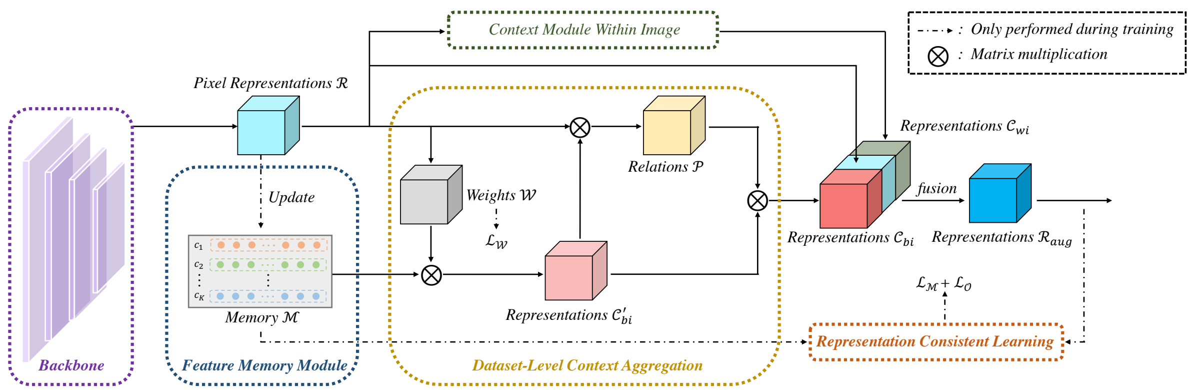

Figure 2. Illustrating the pipeline of mining contextual information beyond the input image. The *Context Module Within Image* denotes for applying existing context scheme within the input image (*e.g.*, PPM [50], ASPP [5] and OCR [45]), which is an optional operation.

Feature Memory Module. Feature memory module enables the deep neural networks to capture the effective information beyond the current input image. It has shown power in various computer vision tasks [9, 25, 33, 40, 41, 55]. For instance, MEGA [9] adopts the memory mechanism to augment the candidate box features of key frame for video object detection. For image search and few shot learning tasks, MNE [25] proposes to leverage memory-based neighbourhood embedding to enhance a general convolutional feature. In deep metric learning, XBM [40] designs a cross-batch memory module to mine informative examples across multiple mini-batches. In this paper, a simple yet effective feature memory module is designed to store the dataset-level representations of various classes so that the network can capture the contextual information beyond the input image and the representation consistent learning is able to be performed. To the best of our knowledge, we are the first to utilize the idea of the feature memory module in semantic image segmentation.

3. Methodology

As illustrated in Figure 2, a backbone network is first utilized to obtain the pixel representations. Then, the pixel representations are used to predict a weight matrix so that the dataset-level representations stored in the feature memory module can be gathered for each pixel. Since the prediction may be inaccurate, the dataset-level representations are further refined by the pixel representations. At last, the original pixel representations are augmented by concatenating the dataset-level representations and thereby, the augmented representations are adopted to predict the final class probability distribution of each pixel.

3.1. Problem Formulation

Given an input image $\mathcal{I} \in \mathbb{R}^{3\times H\times W}$, we first project the pixels in $\mathcal{I}$ into a non-linear embedding space by a backbone network $\mathscr{B}$ as follows:

$$\mathcal{R} = \mathscr{B}(\mathcal{I}), \tag{1}$$

where the matrix $\mathcal{R}$ of size $C \times \frac{H}{8} \times \frac{W}{8}$ stores the pixel representations of $\mathcal{I}$ and the dimension of a pixel representation is C. Then, to mine the contextual information beyond the input image $\mathcal{C}_{bi}$, we have:

$$\mathcal{C}_{bi} = \mathscr{A}_{bi}(\mathscr{H}_1(\mathcal{R}),\ \mathcal{M}), \tag{2}$$

where the dataset-level representations of various categories are stored in the feature memory module $\mathcal{M}$ and $\mathscr{A}_{bi}$ is the proposed dataset-level context aggregation scheme. The matrix $\mathcal{C}_{bi}$ of size $C \times \frac{H}{8} \times \frac{W}{8}$ stores the dataset-level contextual information aggregated from $\mathcal{M}$. $\mathscr{H}_1$ is a classification head used to predict the category probability distribution of the pixel representations in $\mathcal{R}$.

To integrate the proposed method into existing segmentation frameworks seamlessly, we define the self-existing context scheme of the adopted framework as $\mathscr{A}_{wi}$, and then we have:

$$\mathcal{C}_{wi} = \mathscr{A}_{wi}(\mathcal{R}), \tag{3}$$

where $\mathcal{C}_{wi}$ is a matrix storing contextual information within the current input image. Next, $\mathcal{R}$ is augmented as the aggregation of three parts:

$$\mathcal{R}_{aug} = \mathscr{F}(\mathcal{R},\ \mathcal{C}_{bi},\ \mathcal{C}_{wi}), \tag{4}$$

where $\mathscr{F}$ is a transform function used to fuse the original representations $\mathcal{R}$, contextual representations beyond the image $\mathcal{C}_{bi}$ and contextual representations within the image $\mathcal{C}_{wi}$. Note that, $\mathcal{C}_{wi}$ is optional according to the adopted segmentation framework. For instance, if the adopted framework is FCN without any contextual modules, $\mathcal{R}_{aug}$ will be calculated by $\mathscr{F}(\mathcal{R},\ \mathcal{C}_{bi})$. Finally, $\mathcal{R}_{aug}$ is leveraged to predict the label of each pixel in the input image:

$$\mathcal{O} = Upsample_{8\times}(\mathscr{H}_2(\mathcal{R}_{aug})), \tag{5}$$

where $\mathscr{H}_2$ is a classification head and $\mathcal{O}$ is a matrix of size $K \times H \times W$ storing the predicted class probability distribution. K is the number of the classes.

3.2. Feature Memory Module

As illustrated in Figure 2, feature memory module $\mathcal{M}$ of size $K \times C$ is introduced to store the dataset-level representations of various classes. During training, we first randomly select one pixel representation for each category from the train set to initialize $\mathcal{M}$, where the representations are calculated by leveraging $\mathscr{B}$. Then, the values in $\mathcal{M}$ are updated by leveraging moving average after each training iteration:

$$\mathcal{M}_t = (1 - m_{t-1}) \cdot \mathcal{M}_{t-1} + m_{t-1} \cdot \mathscr{T}(\mathcal{R}_{t-1}), \quad (6)$$

where m is the momentum, t denotes for the current number of iterations and $\mathscr{T}$ is used to transform $\mathcal{R}$ to have the same size as $\mathcal{M}$. For m, we adopt polynomial annealing policy to schedule it:

$$m_t = (1 - \frac{t}{T})^p \cdot (m_0 - \frac{m_0}{100}) + \frac{m_0}{100},\ t \in [0, T], \quad (7)$$

where T is the total number of iterations of training. By default, both p and m_0 are set as 0.9 empirically.

To implement $\mathscr{T}$, we first setup a matrix $\mathcal{R}'$ of size $K \times C$ and initialize it by using the values in $\mathcal{M}$. For the convenience of presentation, we leverage the subscript $[i, j]$ or $[i, *]$ to index the element or elements of a matrix. $\mathcal{R}$ is upsampled and permuted as size $HW \times C$, *i.e.*, $\mathcal{R}^{HW \times C}$. Subsequently, for each category c_k existing in $\mathcal{I}$, we have:

$$\mathcal{R}_{c_k} = \{\mathcal{R}^{HW \times C}_{[i,*]} \mid (\mathcal{GT}_{[i]} = c_k) \wedge (1 \leq i \leq HW)\}, \quad (8)$$

where $\mathcal{GT}$ of size HW stores the ground truth category labels of $\mathcal{R}^{HW \times C}$. $\mathcal{R}_{c_k}$ of size $N_{c_k} \times C$ stores the representations of category c_k of $\mathcal{R}^{HW \times C}$. N_{c_k} is the number of pixels labeled as c_k in $\mathcal{I}$. Next, we calculate the cosine similarity matrix $\mathcal{S}_{c_k}$ of size N_{c_k} between $\mathcal{R}_{c_k}$ and $\mathcal{M}_{[c_k,*]}$:

$$\mathcal{S}_{c_k} = \frac{\mathcal{R}_{c_k} \cdot \mathcal{M}_{[c_k,*]}}{\|\mathcal{R}_{c_k}\|_2 \cdot \|\mathcal{M}_{[c_k,*]}\|_2}. \quad (9)$$

Finally, the representation of c_k in $\mathcal{R}'$ is updated as:

$$\mathcal{R}'_{[c_k,*]} = \sum_{i=1}^{N_{c_k}} \frac{1 - S_{c_k,[i]}}{\sum_{j=1}^{N_{c_k}} (1 - S_{c_k,[j]})} \cdot \mathcal{R}_{c_k,[i,*]}. \quad (10)$$

The output of $\mathscr{T}$ is $\mathcal{R}'$ which has been updated by all the representations of various classes in $\mathcal{R}^{HW \times C}$.

3.3. Dataset-Level Context Aggregation

Here, we elaborate the dataset-level context aggregation scheme $\mathscr{A}_{bi}$ to adaptively aggregate dataset-level representations stored in $\mathcal{M}$ for $\mathcal{R}$. As demonstrated in Figure 2, we first predict a weight matrix $\mathcal{W}$ of size $K \times \frac{H}{8} \times \frac{W}{8}$ according to $\mathcal{R}$:

$$\mathcal{W} = \mathscr{H}_1(\mathcal{R}), \quad (11)$$

where $\mathcal{W}$ stores the category probability distribution of each representation in $\mathcal{R}$. The classification head $\mathscr{H}_1$ is implemented by two 1×1 convolutional layers and the $Softmax$ function. Next, we can calculate a coarse dataset-level representation matrix C'_{bi} as follows:

$$\mathcal{C}'_{bi} = permute(\mathcal{W}) \otimes \mathcal{M}, \quad (12)$$

where $\mathcal{C}'_{bi}$ of size $\frac{HW}{64} \times C$ stores the aggregated dataset-level representations for the pixel representations in $\mathcal{R}$. $permute(\mathcal{W})$ is used to make $\mathcal{W}$ have size of $\frac{HW}{64} \times K$ and $\otimes$ denotes for the matrix multiplication. Since $\mathscr{H}_1$ only leverages $\mathcal{R}$ to predict $\mathcal{W}$, the pixel representations may be misclassified. To address this, we calculate the relations between $\mathcal{R}$ and $\mathcal{C}'_{bi}$ so that we can obtain a position confidence weight to further refine $\mathcal{C}'_{bi}$. Specifically, we first calculate the relations $\mathcal{P}$ as follows:

$$\mathcal{P} = Softmax(\frac{g_q(permute(\mathcal{R})) \otimes g_k(\mathcal{C}'_{bi})^T}{\sqrt{\frac{C}{2}}}), \quad (13)$$

where $permute$ is adopted to let $\mathcal{R}$ have size of $\frac{HW}{64} \times C$. Then, $\mathcal{C}'_{bi}$ is refined as follows:

$$\mathcal{C}_{bi} = permute(g_o(\mathcal{P} \otimes g_v(\mathcal{C}'_{bi}))), \quad (14)$$

where g_q, g_k, g_v and g_o are introduced to adjust the dimension of each pixel representation, implemented by a 1×1 convolutional layer. $permute$ is used to let the output have size of $C \times \frac{H}{8} \times \frac{W}{8}$.

3.4. Representation Consistent Learning

Segmentation model based on deep neural networks essentially groups the representations of the pixels of the whole dataset in a non-linear embedding space, while the deep architectures are typically trained by a mini-batch of the dataset at each iteration. Although this can prevent the models from overfitting in a sense, the inconsistency makes the network lack the ability of addressing intra-class compactness and inter-class dispersion from the perspective of the whole dataset. To alleviate this problem, we propose a representation consistent learning strategy.

Specifically, during training, $\mathscr{H}_2$ is also leveraged to predict the labels of the dataset-level representations in $\mathcal{M}$:

$$\mathcal{O}^{\mathcal{M}} = \mathscr{H}_2(reshape(\mathcal{M})), \quad (15)$$

where $reshape$ is utilized to make $\mathcal{M}$ have size of $K \times C \times 1 \times 1$. The classification head $\mathscr{H}_2$ is implemented by two 1×1 convolutional layers and the $Softmax$ function. $\mathcal{O}^{\mathcal{M}}$ stores the predicted category probability distribution of each dataset-level representation in $\mathcal{M}$. It is clear that each representation in $\mathcal{M}$ is essentially a composite vector of the pixel representations of the same category of the whole dataset. Therefore, sharing the classification head in predicting $\mathcal{O}$ and $\mathcal{O}^{\mathcal{M}}$ could make $\mathscr{H}_2$ not only 1) address the categorization ability of the pixels within individual images, but also 2) learn to address intra-class compactness and inter-class dispersion from the perspective of the whole dataset more explicitly.

3.5. Loss Function

A multi-task loss function of $\mathcal{W}$, $\mathcal{O}^{\mathcal{M}}$ and $\mathcal{O}$ is used to jointly optimize the model parameters. Specifically, the objective of $\mathcal{W}$ is defined as follows:

$$\mathcal{L}_{\mathcal{W}} = \frac{1}{H \times W} \sum_{i,j} L_{ce}(\mathcal{W}^{K \times H \times W}_{[*,i,j]},\ \S(\mathcal{GT}_{[ij]})), \quad (16)$$

where $\S$ stands for converting the ground truth category label into one-hot format and $\mathcal{W}^{K \times H \times W}$ is the output of $Upsample_{8\times}(\mathcal{W})$. L_{ce} denotes for the *cross entropy loss* and $\sum_{i,j}$ denotes that the summation is calculated over all locations on the input image $\mathcal{I}$. For $\mathcal{O}^{\mathcal{M}}$, the loss function is formulated as follows:

$$\mathcal{L}_{\mathcal{M}} = \frac{1}{K} \sum_{k=1}^{K} L_{ce}(\mathcal{O}^{\mathcal{M}}_{[*,c_k,1,1]},\ \S(c_k)). \quad (17)$$

To make $\mathcal{O}$ contain the accurate category probability distribution of each pixel, we define the loss function of $\mathcal{O}$ as follows:

$$\mathcal{L}_{\mathcal{O}} = \frac{1}{H \times W} \sum_{i,j} L_{ce}(\mathcal{O}_{[*,i,j]},\ \S(\mathcal{GT}_{[ij]})). \quad (18)$$

Finally, we formulate the multi-task loss function $\mathcal{L}$ as:

$$\mathcal{L} = \alpha \mathcal{L}_{\mathcal{W}} + \beta \mathcal{L}_{\mathcal{M}} + \mathcal{L}_{\mathcal{O}}, \quad (19)$$

where α and β are the hyper-parameters to balance the three losses. We empirically set $\alpha = 0.4$ and $\beta = 1$ by default. With this joint loss function, the model parameters are learned jointly through back propagation. Note that the values in $\mathcal{M}$ are not learnable through back propagation.

4. Experiments

4.1. Experimental Setup

Datasets. We conduct experiments on four widely-used semantic segmentation benchmarks:

- **ADE20K** [56] is a challenging scene parsing dataset that contains 150 classes and diverse scenes with 1,038 image-level labels. The training, validation, and test sets consist of 20K, 2K, 3K images, respectively.
- **LIP** [18] is tasked for single human parsing. There are 19 semantic classes and 1 background class. The dataset is divided into 30K/10K/10K images for training, validation and testing.
- **Cityscapes** [11] densely annotates 19 object categories in urban scene understanding images. This dataset consists of 5,000 finely annotated images, out of which 2,975 are available for training, 500 for validation and 1,524 for testing.
- **COCO-Stuff** [2] is a challenging scene parsing dataset that provides rich annotations for 91 thing classes and 91 stuff classes. This paper adopts the latest version which includes all 164K images (*i.e.*, 118K for training, 5K for validation, 20K for test-dev and 20K for test-challenge) from MS COCO [29].

Training Settings. Our backbone network is initialized by the weights pre-trained on ImageNet [12], while the remaining layers are randomly initialized. Following previous work [5], the learning rate is decayed according to the "poly" learning rate policy with factor $(1 - \frac{iter}{total\_iter})^{0.9}$. Synchronized batch normalization implemented by pytorch is enabled during training. For data augmentaton, we adopt random scaling, horizontal flipping and color jitter. More specifically, the training settings for different datasets are listed as follows:

- **ADE20K:** We set the initial learning rate as 0.01, weight decay as 0.0005, crop size as 512×512, batch size as 16 and training epochs as 130 by default.
- **LIP:** The initial learning rate is set as 0.01 and the weight decay is 0.0005. We set the crop size of the input image as 473×473 and batch size as 32 by default. Besides, if not specified, the networks are fine-tuned for 150 epochs on the train set.
- **Cityscapes:** The initial learning rate is set as 0.01 and the weight decay is 0.0005. We randomly crop out high-resolution patches 512×1024 from the original images as the inputs for training. The batch size and training epochs are set as 8 and 220, respectively.
- **COCO-Stuff:** We set the initial learning rate as 0.01, weight decay as 0.0005, crop size as 512×512, batch size as 16 and training epochs as 30 by default.

Inference Settings. For ADE20K, COCO-Stuff and LIP, the size of the input image during testing is the same as the size of the input image during training. And for Cityscapes, the input image is resized to have its shorter side being 1024 pixels. By default, no tricks (*e.g.*, multi-scale with flipping testing) will be used during testing.

Evaluation Metrics. Following the standard setting, mean intersection-over-union (mIoU) is adopted for evaluation.

Reproducibility. Our method is implemented in PyTorch ($version \geq 1.3$) [34] and trained on 8 NVIDIA Tesla V100 GPUs with a 32 GB memory per-card. And all the testing procedures are performed on a single NVIDIA Tesla V100 GPU. To provide full details of our framework, our code will be made publicly available.

Table 1. Ablation study on the validation set of ADE20K about the form of the transform function in Equation 4. FCN [32] is adopted as the *Baseline* framework.

Method	Backbone	$\mathscr{F}$	mIoU
Baseline	ResNet-50	-	36.96
Baseline+MCIBI (*ours*)	ResNet-50	*add*	40.99
Baseline+MCIBI (*ours*)	ResNet-50	*weighted add*	41.17
Baseline+MCIBI (*ours*)	ResNet-50	*concatenation*	**42.18**

Table 2. Ablation study on the validation set of ADE20K about the proposed feature memory module. "Clustering" denotes that each dataset-level representation of the feature memory module is initialized by clustering of the pixel representations of corresponding category. "RCL" stands for representation consistent learning.

Cosine	Poly	Clustering	RCL	mIoU
✓				41.77
	✓			41.95
✓	✓			42.18
✓	✓	✓		42.20
✓	✓		✓	**42.84**
✓	✓	✓	✓	42.75

4.2. Ablation Study

Transform function in Equation 4. Table 1 shows three fusion methods to fuse $\mathcal{R}$ and C_{bi}. *add* denotes for element-wise adding $\mathcal{R}$ and C_{bi}, which brings 4.03% mIoU improvements. *weighted add* means adding $\mathcal{R}$ and C_{bi} weightedly where the weights are predicted by a 1×1 convolutional layer. It achieves 4.21% mIoU improvements. *concatenation*, which stands for concatenating $\mathcal{R}$ and C_{bi}, is the best choice to implement $\mathscr{F}$. It is 5.22% mIoU higher than *Baseline*. These results together indicate that mining contextual information beyond the input image could effectively improve the pixel representations so that the models could classify the pixels more accurately.

Update Strategy of $\mathcal{M}$. The feature memory module $\mathcal{M}$ is updated by employing moving average, where the current representations of the same category are combined by calculating the cosine similarity and the momentum is adjusted by using polynomial annealing policy. To show the effectiveness, we remove the step of calculating the cosine similarity and leveraging polynomial annealing policy, respectively. As indicated in Table 2, applying cosine similarity and poly policy could bring 0.41% and 0.23% mIoU improvements, respectively.

Initialization Method of $\mathcal{M}$. By default, each dataset-level representation in $\mathcal{M}$ is initialized by randomly selecting a pixel representation of corresponding category. To test the impact of the initialization method on segmentation performance, we also attempt to initialize $\mathcal{M}$ by cosine similarity clustering. Table 2 shows the result, which is very comparable with random initialization (42.18% *v.s.* 42.20%). This result indicates that the proposed update strategy could make $\mathcal{M}$ converge well, without relying on a strict initialization, which shows the robustness of our method.

Table 3. Complexity comparison with existing context schemes. The feature map of size $[1 \times 2048 \times 128 \times 128]$ is used to evaluate their complexity during inference. All the numbers are obtained on a single NVIDIA Tesla V100 GPU with CUDA 11.0 and the smaller, the better.

Method	Parameters	FLOPs	Time
ASPP [6] (*our impl.*)	42.21M	674.47G	101.44ms
PPM [50] (*our impl.*)	23.07M	309.45G	29.57ms
CCNet [22] (*our impl.*)	23.92M	397.38G	56.90ms
DANet [15] (*our impl.*)	23.92M	392.02G	62.64ms
ANN [57] (*our impl.*)	20.32M	335.24G	49.66ms
DNL [43] (*our impl.*)	24.12M	395.25G	68.62ms
APCNet [20] (*our impl.*)	30.46M	413.12G	54.20ms
DCA (*ours*)	14.82M	242.80G	47.64ms
ASPP+DCA (*ours*)	45.63M	730.56G	125.03ms

Table 4. Generalization ability of our feature memory module.

Method	Train Set	Test Set	mIoU
FCN [32]	LIP train set	LIP val set	48.63
DeepLabV3 [6]	LIP train set	LIP val set	52.35
FCN [32]	LIP train set	CIHP val set	27.20
DeepLabV3 [6]	LIP train set	CIHP val set	27.02
FCN+MCIBI (*ours*)	LIP train set	LIP val set	**51.07**
DeepLabV3+MCIBI (*ours*)	LIP train set	LIP val set	**53.52**
FCN+MCIBI (*ours*)	LIP train set	CIHP val set	**27.57**
DeepLabV3+MCIBI (*ours*)	LIP train set	CIHP val set	**27.89**

Representation Consistent Learning. The representation consistent learning strategy (RCL) is designed to make the network learn to address intra-class compactness and inter-class dispersion from the perspective of the whole dataset more explicitly. As illustrated in Table 2, we can see that RCL brings 0.66% improvements on mIoU, which demonstrates the effectiveness of our method.

Complexity Comparison. Table 3 shows the efficiency of our dataset-level context aggregation (DCA) scheme. Compared to the existing context schemes, our DCA requires the least parameters and the least computation complexity, which indicates that the proposed context module is light. Since our context scheme is complementary to the existing context schemes, this result well demonstrates that the model complexity is still tolerable after integrating the proposed module into existing segmentation framework. For instance, after integrating DCA into ASPP, the parameters, FLOPs and inference time is just increased by 2.5M, 40.94G and 23.59ms, respectively.

Generalization Ability. Since the feature memory module stores the dataset-level contextual information to augment the pixel representation, this module may cause the segmentation framework to overfit on the current dataset. To dispel this worry, we train the base model and the proposed method on the train set of LIP, and then test them on the validation set of both LIP and CIHP [17], where CIHP is a much more challenging multi-human parsing dataset with the same categories as LIP. As demonstrated in Table 4, we can see that the dataset-level representations of LIP benefit the model on both LIP (51.07% *v.s.* 48.63% and 53.52% *v.s.* 52.35%) and CIHP (27.57% *v.s.* 27.20% and 27.89%

Table 5. The improvements of segmentation performance achieved on different benchmarks when integrating the proposed mining contextual information beyond image (MCIBI) into the existing segmentation frameworks. All the results here are obtained under single-scale testing without any tricks.

Method	Backbone	Stride	ADE20K (*train/val*)	Cityscapes (*train/val*)	LIP (*train/val*)	COCO-Stuff (*train/val*)
FCN [32]	ResNet-50	8×	36.96	75.16	48.63	35.11
FCN+MCIBI (*ours*)	ResNet-50	8×	**42.84**	**78.50**	**51.07**	**39.95**
PSPNet [50]	ResNet-50	8×	42.64	79.05	51.94	41.01
PSPNet+MCIBI (*ours*)	ResNet-50	8×	**43.77**	**79.90**	**52.84**	**41.80**
OCRNet [45]	ResNet-50	8×	42.47	79.40	52.25	40.57
OCRNet+MCIBI (*ours*)	ResNet-50	8×	**43.34**	**79.98**	**52.80**	**41.10**
DeepLabV3 [6]	ResNet-50	8×	43.19	80.57	52.35	41.86
DeepLabV3+MCIBI (*ours*)	ResNet-50	8×	**44.34**	**81.14**	**53.52**	**42.08**

Table 6. Comparison with the state-of-the-art on ADE20K (val). Multi-scale and flipping testing is adopted for fair comparison.

Method	Backbone	Stride	mIoU
CCNet [22]	ResNet-101	8×	45.76
OCNet [46]	ResNet-101	8×	45.45
ACNet [16]	ResNet-101	8×	45.90
DMNet [19]	ResNet-101	8×	45.50
EncNet [48]	ResNet-101	8×	44.65
PSPNet [50]	ResNet-101	8×	43.29
PSANet [51]	ResNet-101	8×	43.77
APCNet [20]	ResNet-101	8×	45.38
OCRNet [45]	ResNet-101	8×	45.28
PSPNet [50]	ResNet-269	8×	44.94
OCRNet [45]	HRNetV2-W48	4×	45.66
DeepLabV3 [6]	ResNeSt-50	8×	45.12
DeepLabV3 [6]	ResNeSt-101	8×	46.91
SETR-MLA [54]	ViT-Large	16×	50.28
DeepLabV3+MCIBI (*ours*)	ResNet-50	8×	45.95
DeepLabV3+MCIBI (*ours*)	ResNet-101	8×	47.22
DeepLabV3+MCIBI (*ours*)	ResNeSt-101	8×	47.36
DeepLabV3+MCIBI (*ours*)	ViT-Large	16×	**50.80**

v.s. 27.02%). This result well proves that our method owns the generalization ability.

Integrated into Various Segmentation Frameworks. As illustrated in Table 5, we can see that our approach improves the performance of base frameworks (*i.e.*, FCN [32], PSPNet [50], OCRNet [45] and DeepLabV3 [6]) by solid margins. For instance, on ADE20K, mining contextual information beyond image (MCIBI) brings 5.88% mIoU improvements to a simple FCN framework. And for stronger baselines (*i.e.*, PSPNet, OCRNet and DeepLabV3), the performance still gains about 1% mIoU improvements. These results indicate that our method could be seamlessly incorporated into existing segmentation frameworks and brings consistent improvements.

Visualization of Learned Features. As illustrated in Figure 3, we visualize the pixel representations learned by the baseline model (left) and our approach (right), respectively. As seen, after applying our method, the spatial distribution of the learned features are well improved, *i.e.*, the pixel representations with the same category are more compact and the features of various classes are well separated. This result well demonstrates that the dataset-level representations can help augment the pixel representations by improving the spatial distribution of the original output in the non-linear

Table 7. Segmentation results on the validation set of LIP. Flipping testing is leveraged for fair comparison.

Method	Backbone	Stride	mIoU
DeepLab [5]	ResNet-101	-	44.80
CE2P [35]	ResNet-101	16×	53.10
DeepLabV3 [6]	ResNet-101	8×	54.58
OCNet [46]	ResNet-101	8×	54.72
CCNet [22]	ResNet-101	8×	55.47
OCRNet [45]	ResNet-101	8×	55.60
HRNet [38]	HRNetV2-W48	4×	55.90
OCRNet [45]	HRNetV2-W48	4×	56.65
DeepLabV3+MCIBI (*ours*)	ResNet-50	8×	54.08
DeepLabV3+MCIBI (*ours*)	ResNet-101	8×	55.42
DeepLabV3+MCIBI (*ours*)	ResNeSt-101	8×	56.34
DeepLabV3+MCIBI (*ours*)	HRNetV2-W48	4×	**56.99**

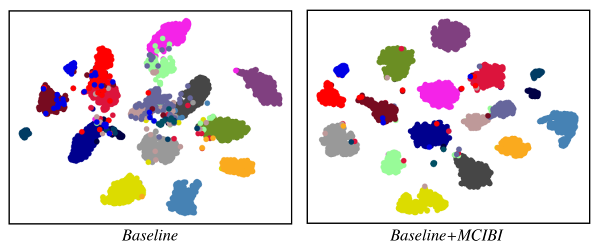

Figure 3. t-SNE visualization on Cityscapes val. A point stands for a composite vector of the pixel representations of the same category within an image. The official recommended color table is used to color the learned features according to their category labels. Obviously, mining contextual information beyond image (MCIBI) can beget a more well-structured semantic feature space.

embedding space so that the network can classify the pixels more smoothly and accurately. More specifically, Figure 4 has illustrated some qualitative results of our method compared to the baseline model.

4.3. Comparison with State-of-the-Art

Results on ADE20K. Table 6 compares the segmentation results on the validation set of ADE20K. Under ResNet-101, ACNet [16] achieves 45.90% mIoU through combining richer local and global contexts, which is the previous best method. By using ResNet-50, our method has achieved a very completive result of 45.95% mIoU. With the same backbone, our approach achieves superior mIoU of 47.22% with a significant margin over ACNet. Prior to this paper, SETR-MLA [54] achieves the state-of-the-art with 50.28%

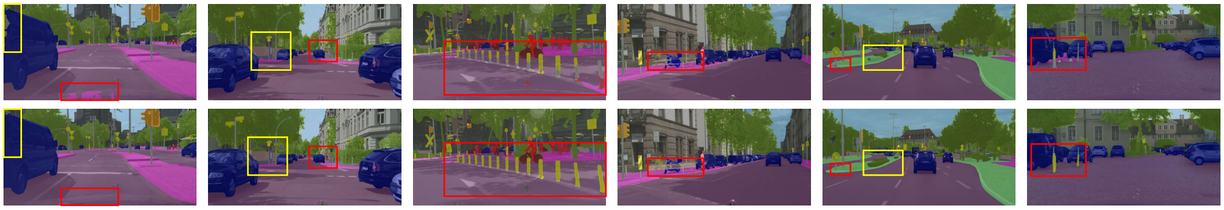

Figure 4. Visual comparisons between FCN (*baseline*) and FCN + MCIBI (*ours*) on the validation set of Cityscapes.

Table 8. Comparison of performance on the test set of Cityscapes with state-of-the-art (trained on trainval set). Multi-scale and flipping testing is used for fair comparison.

Method	Backbone	Stride	mIoU
CCNet [22]	ResNet-101	8×	81.90
PSPNet [50]	ResNet-101	8×	78.40
PSANet [51]	ResNet-101	8×	80.10
OCNet [46]	ResNet-101	8×	80.10
OCRNet [45]	ResNet-101	8×	81.80
DANet [15]	ResNet-101	8×	81.50
ACFNet [47]	ResNet-101	8×	81.80
ANNet [57]	ResNet-101	8×	81.30
ACNet [16]	ResNet-101	8×	82.30
HRNet [38]	HRNetV2-W48	4×	81.60
OCRNet [45]	HRNetV2-W48	4×	82.40
DeepLabV3+MCIBI (*ours*)	ResNet-50	8×	79.90
DeepLabV3+MCIBI (*ours*)	ResNet-101	8×	82.03
DeepLabV3+MCIBI (*ours*)	ResNeSt-101	8×	81.59
DeepLabV3+MCIBI (*ours*)	HRNetV2-W48	4×	**82.55**

mIoU by adopting stronger backbone ViT-Large. Based on this, our method DeepLabV3+MCIBI with ViT-Large achieves a new state-of-the-art with mIoU hitting 50.80%. As is known, ADE20K is very challenging due to its various image sizes, a great deal of semantic categories and the gap between its training and validation set. Therefore, these results can strongly demonstrate the significance of mining contextual information beyond the input image.

Results on LIP. Table 7 has shown the comparison results on the validation set of LIP. It can be seen that our DeepLabV3+MCIBI using ResNeSt101 yields an mIoU of 56.34%, which is among the state-of-the-art. Furthermore, our method achieves a new state-of-the-art with mIoU hitting 56.99% based on the high-resolution network HRNetV2-W48 that is more robust than ResNeSt in semantic image segmentation. This is particularly impressive considering the fact that human parsing is a more challenging fine-grained semantic segmentation task.

Results on Cityscapes. Table 8 demonstrates the comparison results on the test set of Cityscapes. The previous state-of-the-art method OCRNet with HRNetV2-W48 achieves 82.40% mIoU. Thus, we also integrate the proposed MCIBI into HRNetV2-W48 so that we could report a new state-of-the-art performance of 82.55% mIoU.

Results on COCO-Stuff. Since previous methods only report segmentation performance trained on the outdated version of COCO-Stuff (*i.e.*, COCO-Stuff-10K), our method is also only trained on this version for fair comparison. Table 9 reports the performance comparison of our method against six competitors on the test set of COCO-Stuff-10K. As we can see that our DeepLabV3+MCIBI adopting ResNeSt101 achieves an mIoU of 42.15%, outperforms previous best method (*i.e.*, OCRNet leveraging HRNetV2-W48) by 1.65% mIoU. Furthermore, it is also noteworthy that our DeepLabV3+MCIBI using ResNet-101 are also superior to previous best OCRNet even when it uses a more robust backbone HRNetV2-W48. This result firmly suggests the superiority of mining the contextual information beyond the input image. Besides, by leveraging ViT-Large, this paper also reports a new state-of-the-art performance on the test set of COCO-Stuff-10K, *i.e.*, 44.89%

Table 9. Segmentation results on the test set of COCO-Stuff-10K. Multi-scale and flipping testing is employed for fair comparison.

Method	Backbone	Stride	mIoU
DANet [15]	ResNet-101	8×	39.70
OCRNet [45]	ResNet-101	8×	39.50
SVCNet [14]	ResNet-101	8×	39.60
EMANet [26]	ResNet-101	8×	39.90
ACNet [16]	ResNet-101	8×	40.10
OCRNet [45]	HRNetV2-W48	4×	40.50
DeepLabV3+MCIBI (*ours*)	ResNet-50	8×	39.68
DeepLabV3+MCIBI (*ours*)	ResNet-101	8×	41.49
DeepLabV3+MCIBI (*ours*)	ResNeSt-101	8×	42.15
DeepLabV3+MCIBI (*ours*)	ViT-Large	16×	**44.89**

5. Conclusion

This paper presents an alternative perspective for addressing the context aggregation problem in semantic image segmentation. Specifically, we propose to mine the contextual information beyond the input image to further improve the pixel representations. First, we set up a feature memory module to store the dataset-level contextual information of various categories. Second, a dataset-level context aggregation scheme is designed to aggregate the dataset-level representations for each pixel according to their class probability distribution. At last, the aggregated dataset-level contextual information is leveraged to augment the original pixel representations. Furthermore, the dataset-level representations are also used to train the classification head so that we could address intra-class compactness and inter-class dispersion from the perspective of the whole dataset. Extensive experiments demonstrate the effectiveness of our method.

References

[1] Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4413–4421, 2018.

[2] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1209–1218, 2018.

[3] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[6] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[7] Liang-Chieh Chen, Yi Yang, Jiang Wang, Wei Xu, and Alan L Yuille. Attention to scale: Scale-aware semantic image segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3640–3649, 2016.

[8] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[9] Yihong Chen, Yue Cao, Han Hu, and Liwei Wang. Memory enhanced global-local aggregation for video object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10337–10346, 2020.

[10] Ho Kei Cheng, Jihoon Chung, Yu-Wing Tai, and Chi-Keung Tang. Cascadepsp: Toward class-agnostic and very high-resolution segmentation via global and local refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8890–8899, 2020.

[11] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[13] Philipe Ambrozio Dias and Henry Medeiros. Semantic segmentation refinement by monte carlo region growing of high confidence detections. In *Asian Conference on Computer Vision*, pages 131–146. Springer, 2018.

[14] Henghui Ding, Xudong Jiang, Bing Shuai, Ai Qun Liu, and Gang Wang. Semantic correlation promoted shape-variant context for segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8885–8894, 2019.

[15] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019.

[16] Jun Fu, Jing Liu, Yuhang Wang, Yong Li, Yongjun Bao, Jinhui Tang, and Hanqing Lu. Adaptive context network for scene parsing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6748–6757, 2019.

[17] Ke Gong, Xiaodan Liang, Yicheng Li, Yimin Chen, Ming Yang, and Liang Lin. Instance-level human parsing via part grouping network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 770–785, 2018.

[18] Ke Gong, Xiaodan Liang, Dongyu Zhang, Xiaohui Shen, and Liang Lin. Look into person: Self-supervised structure-sensitive learning and a new benchmark for human parsing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 932–940, 2017.

[19] Junjun He, Zhongying Deng, and Yu Qiao. Dynamic multi-scale filters for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3562–3572, 2019.

[20] Junjun He, Zhongying Deng, Lei Zhou, Yali Wang, and Yu Qiao. Adaptive pyramid context network for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7519–7528, 2019.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[22] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 603–612, 2019.

[23] Tsung-Wei Ke, Jyh-Jing Hwang, Ziwei Liu, and Stella X Yu. Adaptive affinity fields for semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 587–602, 2018.

[24] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6399–6408, 2019.

[25] Suichan Li, Dapeng Chen, Bin Liu, Nenghai Yu, and Rui Zhao. Memory-based neighbourhood embedding for visual

recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6102–6111, 2019.

[26] Xia Li, Zhisheng Zhong, Jianlong Wu, Yibo Yang, Zhouchen Lin, and Hong Liu. Expectation-maximization attention networks for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9167–9176, 2019.

[27] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1925–1934, 2017.

[28] Guosheng Lin, Chunhua Shen, Anton Van Den Hengel, and Ian Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3194–3203, 2016.

[29] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[30] Wei Liu, Andrew Rabinovich, and Alexander C Berg. Parsenet: Looking wider to see better. *arXiv preprint arXiv:1506.04579*, 2015.

[31] Ziwei Liu, Xiaoxiao Li, Ping Luo, Chen Change Loy, and Xiaoou Tang. Deep learning markov random field for semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 40(8):1814–1828, 2017.

[32] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[33] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9226–9235, 2019.

[34] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

[35] Tao Ruan, Ting Liu, Zilong Huang, Yunchao Wei, Shikui Wei, and Yao Zhao. Devil in the details: Towards accurate single and multiple human parsing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4814–4821, 2019.

[36] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[38] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[39] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[40] Xun Wang, Haozhi Zhang, Weilin Huang, and Matthew R Scott. Cross-batch memory for embedding learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6388–6397, 2020.

[41] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 284–293, 2019.

[42] Maoke Yang, Kun Yu, Chi Zhang, Zhiwei Li, and Kuiyuan Yang. Denseaspp for semantic segmentation in street scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3684–3692, 2018.

[43] Minghao Yin, Zhuliang Yao, Yue Cao, Xiu Li, Zheng Zhang, Stephen Lin, and Han Hu. Disentangled non-local neural networks. In *European Conference on Computer Vision*, pages 191–207. Springer, 2020.

[44] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 472–480, 2017.

[45] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. *arXiv preprint arXiv:1909.11065*, 2019.

[46] Yuhui Yuan and Jingdong Wang. Ocnet: Object context network for scene parsing. *arXiv preprint arXiv:1809.00916*, 2018.

[47] Fan Zhang, Yanqin Chen, Zhihang Li, Zhibin Hong, Jingtuo Liu, Feifei Ma, Junyu Han, and Errui Ding. Acfnet: Attentional class feature network for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6798–6807, 2019.

[48] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7151–7160, 2018.

[49] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Zhi Zhang, Haibin Lin, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al. Resnest: Split-attention networks. *arXiv preprint arXiv:2004.08955*, 2020.

[50] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[51] Hengshuang Zhao, Yi Zhang, Shu Liu, Jianping Shi, Chen Change Loy, Dahua Lin, and Jiaya Jia. Psanet: Point-wise spatial attention network for scene parsing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 267–283, 2018.

[52] Shuai Zhao, Yang Wang, Zheng Yang, and Deng Cai. Region mutual information loss for semantic segmentation. *arXiv preprint arXiv:1910.12037*, 2019.

[53] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1529–1537, 2015.

[54] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6881–6890, 2021.

[55] Zhun Zhong, Liang Zheng, Zhiming Luo, Shaozi Li, and Yi Yang. Invariance matters: Exemplar memory for domain adaptive person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 598–607, 2019.

[56] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017.

[57] Zhen Zhu, Mengde Xu, Song Bai, Tengteng Huang, and Xiang Bai. Asymmetric non-local neural networks for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 593–602, 2019.